新しい思考の道具をつくる
— 複雑系による社会のモデル化とシミュレーション —

井庭　崇†

† 慶應義塾大学 総合政策学部

E-mail: †iba@sfc.keio.ac.jp

要旨　本論文では、複雑系のシステム観に基づく社会・経済シミュレーションを作成するための新しい思考の道具を提案する。本論文における複雑系とは、広義には「内部状態をもつ多数の構成要素が相互作用し、それぞれの内部状態を変化させていくシステム」であり、狭義には、これに加えて「構成要素の振舞いのルールが変化し得る」という定義が加わる。近年、社会科学においてこのような捉え方が重要視されているが、現状では、複雑系のモデルを記述し操作するための有効な手段は存在しない。本論文では、複雑系の社会・経済モデルを記述・操作するための支援として、マルチエージェントモデルを導入し、モデル・フレームワーク、モデル作成プロセス、モデル作成支援ツール、シミュレーション・プラットフォームを提案する。

キーワード　複雑系, シミュレーション, マルチエージェントモデル, オブジェクト指向, メタファー

Creating New Tools for Thinking
— Modeling and Simulating Societies as Complex Systems —

Takashi IBA†

† Faculty of Policy Management, Keio University

E-mail: †iba@sfc.keio.ac.jp

Abstract　This paper presents a new tools for thinking for simulating economies and societies as complex systems. In a broad sense, in this paper, the complex system means that the system has the components where each component changes the internal states by mutually interacting with the other components. In addition, in a strict sense, the complex system means that the rules of each component behavior are changed dynamically over time. There is no satisfactory scheme for modeling and simulating the complex systems, although the complex system model has been highly demanded in social sciences. We introduce an agent-based modeling for social sciences in order to model and simulate the complex system where the model framework, modeling process, modeling tools, and simulation platform are proposed.

Key words　complex systems, simulation, multi-agent model, object-orientation, metaphor

1. はじめに

本研究の目的は、社会・経済を分析するための新しい思考の道具を構築することにある。この思考の道具は、「組織化された複雑性」の領域にある社会・経済現象を、複雑系のシステム論的アプローチで取り組むことを支援するものである。社会・経済という組織化された複雑性をもつ対象は、従来の解析的方法や統計的方法では理解が困難であるといわれているが、本研究では、システム論とコンピュータ・シミュレーションによって取り組むことにしたい。

社会・経済システムの研究は、同時代のシステム論からの影響を受けて発展してきており、近年「複雑系」(complex system) と呼ばれるシステム観が重要視されている。ところが現在、「複雑系」として社会・経済を分析するための有効な方法と道具立ては存在しない。このことが、研究の進展を困難なものにしているということ、そして今後その問題がさらに深刻化するということが、本論文の基本的な問題意識である。

本論文では、社会・経済を「複雑系」として記述・分析するために、私たちがこまで開発してきた「モデル・フレームワーク」と「モデル作成プロセス」、「モデル作成支援ツール」、「シミュレーション・プラットフォーム」を紹介することにしたい。

2. メタファーとしてのモデル

社会・経済を理解したいとき、社会科学では、対象となる現象の説明根拠を、その社会そのもの、あるいはそれを構成する人間に求め、その要因を把握することを試みる。もちろん実際には、「そのような要素は、具体的現象においては相互に入り組んでおり、ほとんどのばあい説明もできなければ把握できないほどに縺れ合っている」[1] ため、本当の意味ですべての因果の連鎖を把握することは不可能である。それゆえ、本質的に重要だと思われる連関についての「モデル」を作成し、現象を理解したり予測したりすることになる (図 1)。

モデルとは何かという定義にはいろいろなものがあるが、ここではWilson [2] による次のような定義を想定しておくことにしよう。「“モデル” とは、ある人間にとっての、ある状況、あるいは状況についての概念 (idea) の明示的な解釈 (explicit interpretation) である。モデルは、数式、記号、あるいは言葉で表すことができるが、本質的には、実体、プロセス、属性、

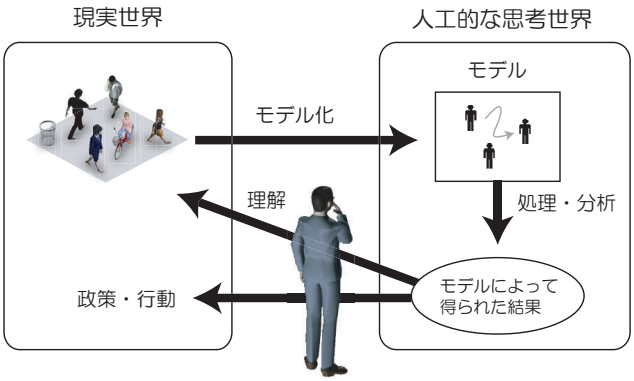

図 1　モデルによる思考

およびそれらの関係についての記述 (description) である」[2]。

このように捉えると、モデルという知的構築物は、メタファー (隠喩)(注1) の役割を果たしていると考えることができる [4], [5]。ここでいうメタファーとは、単なる言葉の綾や修辞的な文飾のことではなく、人間の認知や思考に組み込まれた「見立て」の方法のことである [6]。メタファーの基本要素は、「たとえられるもの」と「たとえるもの」、「そのたとえの根拠」であるが、この場合、現実世界における対象が「たとえられるもの」、モデルが「たとえるもの」である。科学的研究とは、「たとえるもの」(モデル) を作成し、「そのたとえの根拠」を、実験などを通じて検証・確証・反証していくという営みだということができる。

近年の科学哲学では、科学的知識といえども客観的なものではなく、それぞれの科学者の認識の枠組みで解釈され構成されたものであるといわれている [7], [8]。観察行為というプリミティヴな行為でさえ、現実からありのままの「事実」を受動的に受けとっているのではなく、「〜として見る」(seeing as) や「〜ことを見る」(seeing that) というメカニズムが不可避的に組み込まれているのである。そのため、私たちの思考の中の「たとえるもの」の表現力が貧弱であれば、分析によってわかることも貧弱なものにならざるを得ないということになる。「認識装置が新たに開発されてはじめて、既存のそれにはなかったリアリティが取りだせる」[9] ことから、「たとえるもの」の表現力を上げていくことも、科学における

(注1)：メタファーは、物事の類似性を間接的に暗示する。直喩が「A は B のような」というのに対し、メタファー (隠喩) は「A は B である」という形式になる。メタファーは、「より抽象的で分かりにくいカテゴリーに属する対象を、より具体的で分かりやすいカテゴリーに属する対象に見立てることによって、世界をよりよく理解する方法」[3] であるといわれている。

重要な活動となる。

以下では、社会・経済を複雑系のメタファーによって表現し、思考するための新しい道具を構築することを提案する。

3. 複雑系のメタファー

現在のところ「複雑系」という用語について確立された定義や明確な合意があるわけではない。複雑系の定義は、研究者によって、あるいは時代によって、まったく異なる意味で用いられており、また、未定義語のまま使用されていることも多い。

そこで本論文では、混乱した状況を整理して議論しやすくするために、「広義の複雑系」と「狭義の複雑系」という二つの定義を行うことにしたい [10]。第一の定義である「広義の複雑系」とは、「内部状態をもつ構成要素が多数相互作用するシステム」のことである。これに対し、第二の定義である「狭義の複雑系」とは、上記の定義で規定されるシステムの中でも、特に「構成要素の振舞いのルールが動的に変化するシステム」のことを指す。

以下では、両定義に共通する複雑系の特徴として、要素に関する特徴と、階層間の関係についての特徴をみていくことにしたい。

3.1 複雑系における要素

物理学をはじめとして自然科学では、対象をより小さな部分へと分解していくことにより、最終的には不変の最小単位 (アトム) に到達すると考えられてきた (図 2)。原子論的なアトムである構成要素は、全体から切り取られても、切り取る前の性質を保ったままである。それゆえ、対象を要素に還元して理解し、その後、要素を足し合わせて全体を理解するという理解の仕方が可能となる。

これ対し、複雑系の構成要素は、原子論的な意味でのアトムではなく、内部状態をもつという点に特徴がある。内部状態をもつということは、外から決めることのできない内部自由度をもっていることを表している。それゆえ、その振舞いを知るためには、いまどの状態にあるのかということを考慮する必要がでてくる。また、そのために、どのようにその状態に行き着いたのか、という文脈 (コンテクスト) についても、注意を払うことが必要になる。

社会科学では、社会を構成する人間は「物理学的なアトムではない」ということが繰り返し強調されてきた [11]。人間は複数の内部状態をもっており、それらはその人が置かれている状況や役割、体調などの要因によって刻々と変化していく。そして、その

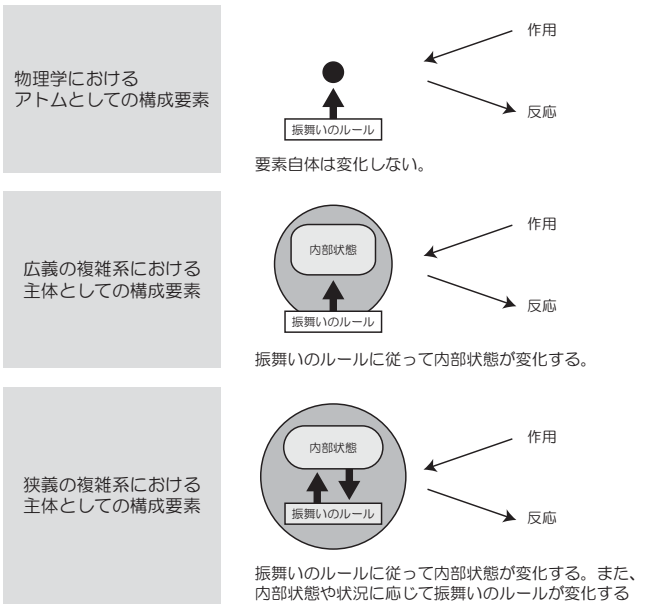

図 2　物理学、広義の複雑系、および狭義の複雑系における構成要素の特徴

内部状態に依存して、価値基準や判断が変化したり、状況の認知や他者との関係が影響を受けるのである。

このように構成要素を捉えることは、内部に自由度をもった主体が、相互作用を行ってその状態を変化させていく点に注目するということである。このため、構成要素は、「自律的」(autonomous) であるといわれる。自律とは、外部からの作用が行なわれたとしても、自分自身の原理で処理することである(注2)。

3.2 上位から下位の層への影響

複雑系では、「システム階層における上位の層から下位の層への影響」があると捉えられることが多い(注3)。つまり、下位の層の構成要素の振舞いや状態が、上位の層の状況によって変化するということである (図 3)。これは従来のシステム観にはない特徴であり、近年ミクロ・マクロ・ループという概念で議論されているものである [13]。上位の層は下位の層を前提とするが、逆に下位の層も上位の層に依存していることになり、その点にこそ、複雑系が還元的手法では理解できない理由がある [14]。

(注2)：これに対し、他律とは、外部から他の原理が持ち込まれ、それによって動かされるということである。

(注3)：複雑系に関する議論のなかで、「複雑系とは創発が起こるシステムである」といわれることがある。創発とは、システムの下位の層では見られなかった特性が、上位の層で現れることであり、秩序形成を考える上で重要な概念である。ところが、たしかに創発の概念は重要であるものの、それだけでは複雑系の定義としては十分とはいえないだろう。なぜなら、複雑系以前のシステム論においても、システムの階層性や創発については言及されており、創発自体は複雑系に特有の現象ではなく、システム一般の特徴だといえるからである。複雑系という新しい概念をわざわざつくるのであれば、「複雑系だとあえて呼ばなければならない必

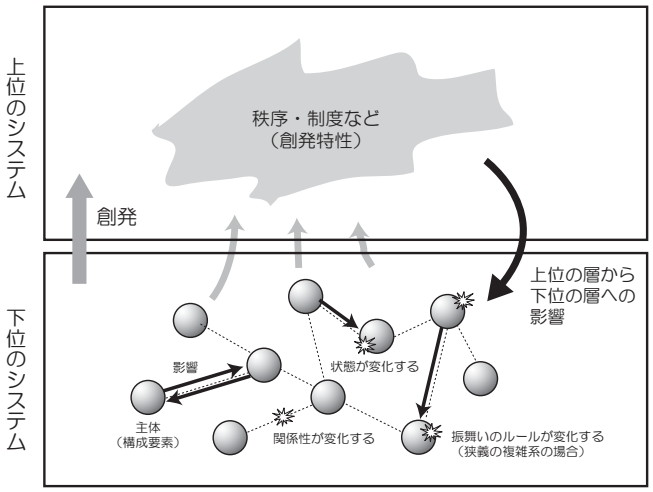

図 3　複雑系における階層間の関係性

4.　モデル・フレームワーク

本論文でこれまでに述べてきたような複雑系のモデルの記述と分析を可能とするために、マルチエージェントモデルによるモデル化を採用することにしたい。マルチエージェントモデルとは、複数の異なる種類のエージェント (自律的主体) が相互作用するというモデルであり、近年コンピュータ・シミュレーションによってその研究が進められているものである。

ここでは、「広義の複雑系」および「狭義の複雑系」の捉え方で社会・経済を記述するためのモデル・フレームワークとして、「Boxed Economy 基礎モデル」(以下、基礎モデル) を提案する(注4)。基礎モデルは、社会・経済をモデル化する際に共通して登場する要素と構造をオブジェクト指向によって抽象化し、定義したものである。以下では、基礎モデルを静的な側面と動的な側面に分けて説明した後、複雑系のモデルの表現方法について考察する。

4.1　静的な側面

基礎モデルの中心的な部分は、World、Space、Clock、Entity、Agent、Goods、Information、Behavior、Relation、Channel というクラスで構成されている (図 4)。まず、対象世界を表現する土台が「World」である。世界は、その世界に固有の空間と時間によって規定されており、それが「Space」と「Clock」で表される。この「Clock」の時間が進むことで、シミュレーション上の時間が経過する(注5)。

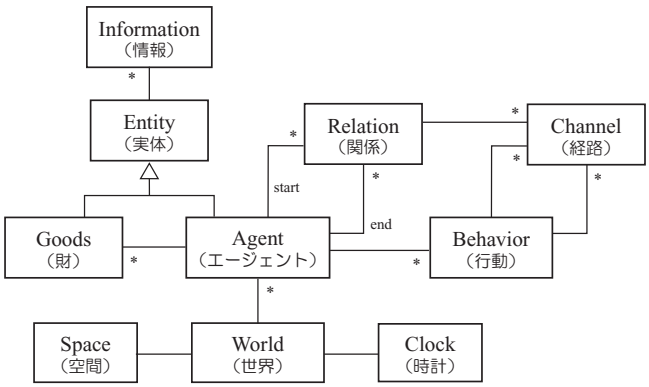

図 4　Boxed Economy 基礎モデル

世界には「Entity」、すなわち「Agent」と「Goods」が複数存在する。Agent とは、社会・経済においてさまざまな活動を行う個人や社会集団 (企業・政府・家族・学校・地域社会・国) のことである。それぞれの Agent には、個性をもたせ、多様な振舞いや状態をとらせることができる。Goods は、Agent に所有し交換される有形/無形の「もの」である。ここでいう「Goods」とは、人間の欲求を充足する性質をもつという経済学における狭義の意味ではなく、世界におけるさまざまなものを示す広義の概念として用いられている。例えば、自動車、石油、トウモロコシ、株、土地の権利、広告、書籍、水、声、騒音、ごみ、貨幣などは、どれも Goods オブジェクトとして表される。そして、Agent が記憶した情報や、Goods に付随して取引される情報などは、「Information」として表される。

Agent の行動は、「Behavior」として表現する。例えば、企業における生産行動や販売行動、個人における購買行動や労働行動などはどれも Behavior であり、これらの Behavior によって、Goods や Information の作成や処理を行う。Agent は複数の Behavior をもつことができ、それらを並列的に実行することができる。

ある Agent から他の Agent への関連性は、「Relation」によって表現する。これにより、友人関係や家族関係、雇用関係などの関係性を表現することができる。実際のコミュニケーションの際には、この Relation に基づいて開設されるコミュニケーション・パスである「Channel」を通じて、商品や会話、貨幣などの Goods と Information のやりとりを行う(注6)。

然性」[12] があるべきだ、という主張に賛成である。
(注4)：「Boxed Economy」という名称は、ブラックボックスとしての経済社会モデルを開けてみると、そこに「箱詰めされた経済」がある、というイメージから命名したものである。もともとは経済分野に注目して開発していたため、「Economy」という言葉が使われているが、現在では経済だけでなく、それ以外の社会的な現象にも適用できるようになっている。

(注5)：Clock の時間が進むと、後に説明する「TimeEvent」が発生する。
(注6)：Channel を通じて Goods や Information が送られてくると、後に説明する「ChannelEvent」が発生する。

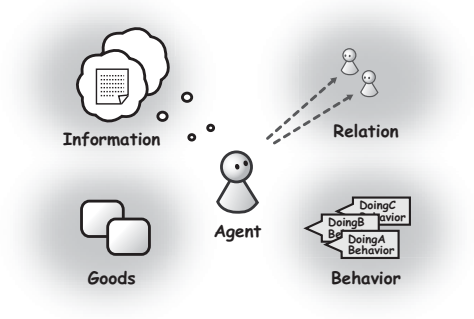

図 5　Agent は Behavior・Relation・Goods・Information をもつ

以上のように、基礎モデルでは、エージェントのもつ行動・関係・財・情報を外部化し、Agent オブジェクトに付加するという方法を採用している(注7)。これらの外部化された要素を組み合わせることによって、エージェントを設定していくのである (図 5)。そのため、Agent 自体はそれらを束ねる役割を果たしているにすぎないということになる。例えば、Agent をインスタンス化すると、単なる Agent オブジェクトが得られるが、そこに PurchaseBehavior を加えると、購買行動を行う「消費者エージェント」となる。このようなエージェントの設計によって、新しい行動・関係・財・情報の追加や削除、組み換えなどを動的に行うことが可能になるのである [17], [18]。このようなエージェントの設計が、このモデル・フレームワークの最大の特徴といえる。

4.2　動的な側面

基礎モデルでは、Behavior のひとつひとつを「状態機械」(オートマトン) として記述する。状態機械とは、トリガーとなるイベント (影響を及ぼすさまざまな出来事) を受け取ると、現在の状態に応じたアクション (動作) を行い、次の状態へ遷移するというシステムである。Behavior の状態遷移を引き起こすイベントには、時間が経過したことを表す「TimeEvent」と、Goods や Information が送られてきたことを表す「ChannelEvent」がある。つまり、エージェントの Behavior は、時間が経過した場合か、他のエージェントから何らかの働きかけがあった場合に活性化することになる。このような状態機械のすべての状態の見取り図は、状態遷移図を用いて表現することができる。

(注7)：このような設計を、オブジェクト指向の分野では、「オブジェクトコンポジション」という。オブジェクトコンポジションとは、役割を外部化するためのオブジェクトを用意して振舞いを委譲し、そのオブジェクトを実行時に関連づける設計のことである [15], [16]。

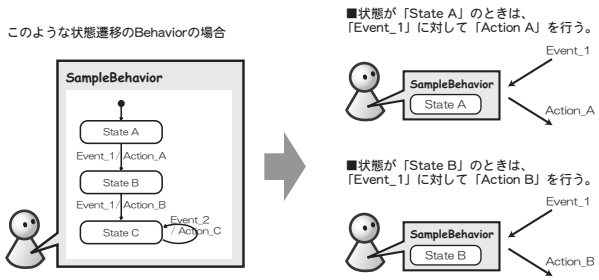

図 6　広義の複雑系における「内部状態に応じた反応」の表現方法

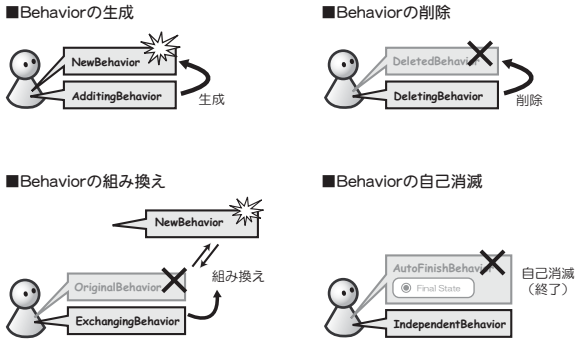

図 7　狭義の複雑系における「振舞いのルールの変化」の表現方法

4.3　複雑系の表現方法

提案したモデル・フレームワークによって、複雑系のモデルがどのように表現できるのかを考えてみよう。まず最初に、内部状態によって反応が異なるという「広義の複雑系」のモデルは、図 6 のように表現できる。このエージェントは、Behavior のそのときの状態によって、同じイベント Event_1 に対して異なる反応をする。つまり、State_A のときには Action_A を行い、State_B のときには Action_B を行うのである。

次に、行動のルールが動的に変化するという「狭義の複雑系」のモデルは、図 7 のように表現できる。エージェントは、Behavior を動的に追加・交換することができ、また、Behavior を削除したり、Behavior の状態遷移が完了すると自動的に消滅するようにすることもできる。これらによって、エージェントの振舞いが変化することになる。

5.　モデル作成プロセス

提案したモデル・フレームワークを用いたモデル作成を支援するため、私たちはモデル作成プロセスを提案している。そのモデル作成プロセスは、「概念モデリング フェーズ」「シミュレーションデザイン フェーズ」「実行・検証フェーズ」のスパイラルモデルになっている [19]。なお、このプロセスにおける

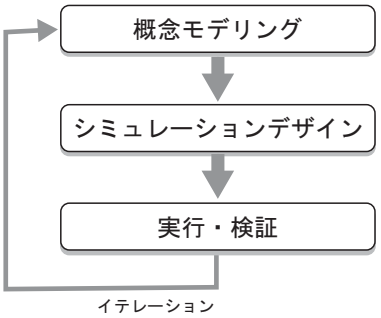

図 8　モデル作成プロセスの概要

ほぼすべての作業は、後述のモデル作成支援ツール「Component Builder」を用いて行うことができる。

5.1　概念モデリング フェーズ

概念モデリング フェーズの目的は、これから作るシミュレーションがどのようなものなのかを明らかにし、概念モデルとして記述することである。その際には、「Boxed Economy 基礎モデル」に基づいて、対象世界を分析し、モデルを記述していく。

まず最初に行うのは、シミュレーションに登場する「エージェント」とその「行動」、エージェント間の「関係」、そして「財」や「情報」をすべて洗い出し、概念モデルクラス図として記述することである。

次に、それらのエージェントがどのように振舞うのかを分析し記述する。各エージェントの行動がどのような手順（フロー）で行われるのかについては、アクティビティ図に記述していく。また、複数のエージェントがどのような順序（シーケンス）でコミュニケートするのかについては、コミュニケーション・シーケンス図に記述していく。以上の作業を、必要に応じて繰り返しながら、概念モデルを作成する。

5.2　シミュレーションデザイン フェーズ

シミュレーションデザイン フェーズの目的は、概念モデルを、プログラムを記述するためのモデルに変換し、実装していくことである。その際に、基礎モデルに基づいたソフトウェア・フレームワークの仕様に合うように作成していく。

まず、概念モデリングフェーズで作成した概念モデルクラス図をもとに、シミュレーションモデルクラス図を作成する。次に、動的な振舞いを記述するために、シミュレーションモデルクラス図、コミュニケーション・シーケンス図、アクティビティ図を参考に、状態遷移図を作成する。

それから、設計したモデルをコンピュータ上でシミュレートできるプログラムとして実装する。後述するモデル作成支援ツール「Component Builder」を用いることで、ソースコードの大部分は自動生成さ

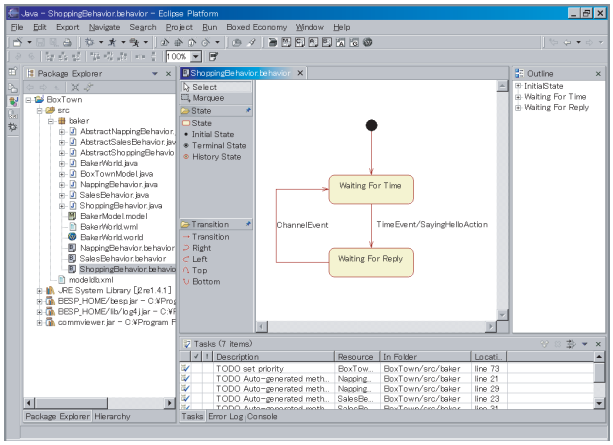

図 9　Component Builder の画面

れるが、現在のバージョンでは、状態遷移中のアクションの部分のプログラムを書く必要がある(注8)。

5.3　実行・検証 フェーズ

実行・検証 フェーズでは、これまで設計・実装してきたモデルをシミュレーション・プラットフォーム上で実行し、その評価を行う。シミュレーションから信頼できる結果を得るためには、モデルの正当性を検証する必要がある。正当性の検証とは、考えていた概念モデルが、きちんとプログラムに変換されたかを検証することである。そのために実行結果やログを利用してモデルの正当性の検証を行う。

6.　モデル作成支援ツール

提案したモデル・フレームワークとモデル作成プロセスによるモデル化を支援するために、私たちは、モデル作成支援ツール「Component Builder」を提案している [21]。Component Builder は、4 種類のデザイナ（エディタ）と、世界設定のためのコンポーザで構成されており、これらを組み合わせて用いることで、シミュレーションのモデル・コンポーネントを作成することができる(注9)。

このツールは、モデル図の記述を支援するとともに、シミュレーションを作成する際のプログラミングを大幅に軽減させる機能も提供されている。モデル図を作成すると、プログラムコードを自動生成してくれるため、プログラミングで難関となりやすい

(注8)：現在、Action 部分を記述するための Action Designer を開発中である。これが完成すると、Java 言語によるソースコードをまったく記述せずにシミュレーション作成ができるようになる [20]。

(注9)：Component Builder は、統合開発環境「eclipse」のプラグインとして開発されている。機能性と使いやすさの面でも定評がある開発環境をベースとすることで、より効率的なシミュレーション作成が可能となる。eclipse については、http://www.eclipse.org/を参照してほしい。

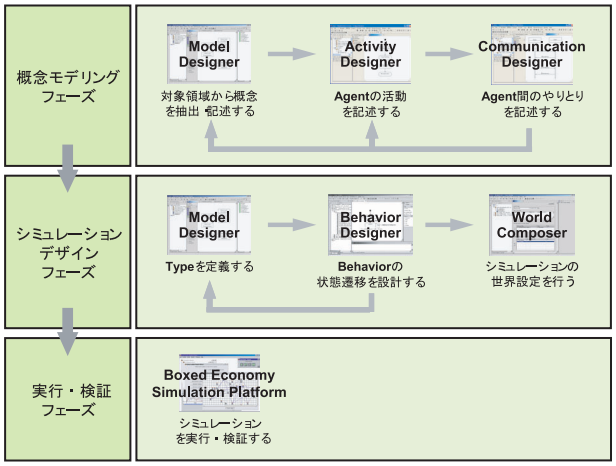

図 10　モデル作成プロセスとその支援ツール

部分をプログラミングしなくて済むので、振る舞いが複雑なモデルであっても比較的容易に作成できるようになる。

6.1　Model Designer

概念モデルクラス図とシミュレーションモデルクラス図の作成には、Model Designer を用いることができる。Model Designer は、UML のクラス図を用いて社会モデルの静的な構造をモデル化するためのエディタであり、記述したモデルから、シミュレーション・プラットフォーム上で動作するコンポーネントのソースコードを生成することができる。

6.2　Activity Designer

アクティビティ図の作成には、Activity Designer を用いることができる。Activity Designer は、UML のアクテビティ図を用いて Agent の活動をモデル化するエディタである。

6.3　Communication Designer

コミュニケーション・シーケンス図の作成には、Communication Designer を用いることができる。Communication Designer は、UML のシーケンス図に準じた図を用いて、Agent 間のやり取りをモデル化するエディタである。

6.4　Behavior Designer

状態遷移図の作成には、Behavior Designer を用いることができる。Behavior Designer は、UML のステートチャート図を用いて、Behavior の状態遷移をモデル化するエディタである。モデル化した Behavior の状態遷移から、シミュレーション・プラットフォーム上で動作するコンポーネントのソースコードを生成することができる。

6.5　World Composer

世界の設定には、World Composer を用いることができる。World Composer は、シミュレーションの世界の初期状態を記述するためのツールである。GUI を使って初期状態を記述し、そこからシミュレーション・プラットフォーム上で動作するソースコードを生成することができる。

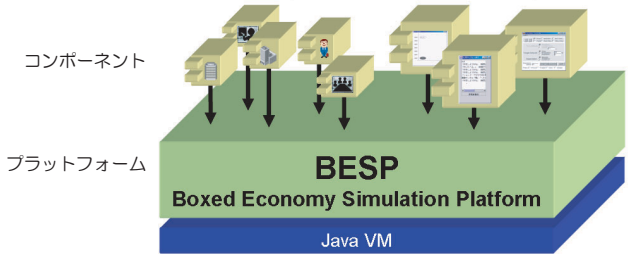

図 11　Boxed Economy Simulation Platform のアーキテクチャ

7.　シミュレーション・プラットフォーム

モデル・フレームワークにもとづくシミュレーションを実行・分析するためのソフトウェアとして、私たちは Boxed Economy Simulation Platform (以下、BESP) を提案している。BESP は、マルチエージェントモデルによる社会・経済のシミュレーションを実行・分析するためのプラットフォームである。

BESP では、シミュレーションプログラムを、個々の機能単位ごとに分割された「コンポーネント」と、それをまとめる「プラットフォーム」からなるシステムとして設計されている。このような設計を採用するのは、モデルや機能をコンポーネントに分割して独立させることで、容易に一部を再利用・拡張したりできるようにするためである。シミュレーションは、複数のコンポーネントの組み合わせによって動作するが、それぞれのコンポーネントは、独立して理解したり作成したりすることができる。今後作成したいモデルが複雑かつ大規模になるにつれて、一つの研究グループでモデルのすべてを作りきれなくなると予想されるため、コンポーネントの再利用性はますます重要になるだろう。このようなコンポーネントによる設計は、「大規模システムをモジュールに分けて考える」という工学の知恵のひとつである。

8.　適用事例の紹介

私たちはこれまで、本論文で提案してきた仕組みを、様々な分野に適用してきた。成長するネットワークモデル [10]、世代交代ネットワークモデル [22]、情報伝播モデル [23]、繰り返し囚人のジレンマにおけ

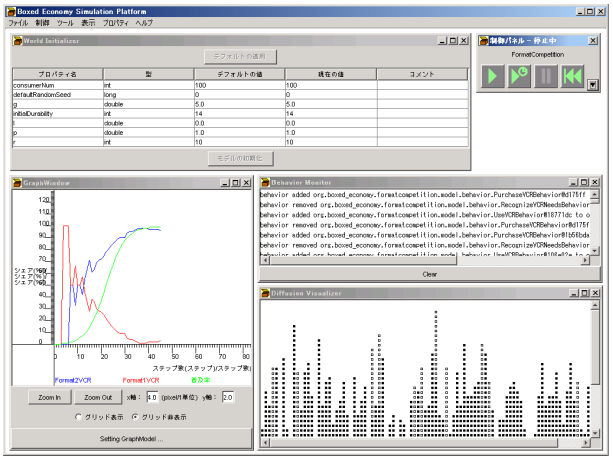

図 12　Boxed Economy Simulation Platform の画面

る戦略模倣 [24]、貨幣の自生と自壊モデル [10]、進化経済モデル、空港の待ち行列モデル、家庭用ビデオの規格競争 [25]、株式市場モデル [26]、電力市場モデル [27] などである。ここでは紙面の都合上、それぞれのモデルについて個別に紹介することはできないので、各論文を参照していただきたい。

9. おわりに

「テクノロジーはわれわれを賢くしてくれる可能性をもっている」が、同時に「われわれを愚かにする可能性もある」[28]———この点は、思考の道具をデザインする際に、強調しすぎてしすぎることはない。本論文の提案において、本質的に重要なのは、プログラミング作業が減るというような技術や効率性の問題ではなく、モデリングが強調され、そこに重点を置くことができるようになるという点にある。私たちが目指しているのは、モデリングとシミュレーションによる思考を支援するということである。そのため、機械に頼って人間の能力が退化してしまような「電卓」ではなく、「習熟すると『答えを生成する仕組み』が頭の中に出来上がる」[29] ような「そろばん」の開発を意識する必要がある。本論文の提案が、複雑系や非線形性を伴う現象の思考について「われわれを賢くしてくれる」道具につながることを切に願う。

文　献

[1] シュムペーター. 社会科学の過去と未来. ダイヤモンド社, 1972.

[2] Brian Wilson. システム仕様の分析学：ソフトシステム方法論. 共立出版, 第 2 版, 1996.

[3] 瀬戸賢一. 空間とレトリック. 海鳴社, 1995.

[4] M. Black. *Models and Metaphors: Studies in Language and Philosophy*. Cornell University Press, 1962.

[5] M. ヘッセ. 科学・モデル・アナロジー. 培風館, 1986.

[6] M. ジョンソン G. レイコフ. レトリックと人生. 大修館書店, 1986.

[7] ノーウッド・ラッセルハンソン. 知覚と発見: 科学的探究の論理. 紀伊国屋書店, 1982.

[8] トーマス・クーン. 科学革命の構造. みすず書房, 1971.

[9] 今田高俊. 自己組織性: 社会理論の復活. 創文社, 1986.

[10] 井庭崇. 社会・経済シミュレーションの基盤構築：複雑系と進化の理論に向けて, 2003.

[11] 村上泰亮. 伝統的思考の宿酔から醒めるとき. 村上泰亮著作集 1. 中央公論社, 1997. 初出：週刊東洋経済臨時増刊 ＜経済体制特集＞, 1967.4, 25 頁.

[12] 金子邦彦, 津田一郎. 複雑系のカオス的シナリオ. 朝倉書店, 1996.

[13] 塩沢由典. ミクロ・マクロ・ループについて. 京都大学経済学会・経済論叢, Vol. 164-5, pp. 463–535, 1999.

[14] 井庭崇, 福原義久. 複雑系入門: 知のフロンティアへの冒険. NTT 出版, 1998.

[15] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. オブジェクト指向における再利用のための デザインパターン. ソフトバンクパブリッシング, 1999.

[16] マーク・メイフィールドピーター・コード. UML による Java オブジェクト設計. ピアソンエデュケーション, 第 2 版, 2000.

[17] 井庭崇. エージェントベース経済シミュレーションのためのエージェント設計論. オペレーションズ・リサーチ 経営の科学, Vol. 46, No. 10, pp. 561 – 567, 2001.

[18] 井庭崇, 中鉢欣秀, 松澤芳昭, 海保研, 武藤佳恭. Boxed economy foundation model: 社会・経済のエージェントベースモデリングのためのフレームワーク. 情報処理学会論文誌: 数理モデル化と応用, Vol. 44, No. SIG14(TOM9), 2003.

[19] Boxed Economy Project. 社会シミュレーション デザイナーズガイド (第 2 版). フジタ未来経営研究所, 2004. http://www.boxed-economy.org/.

[20] N. Aoyama, R. Takeda, T. Iba, and H. Ohiwa. Simulation development tools with mda. In *International Workshop on Massively Multiagent Systems*, 2004.

[21] T. Iba, Y. Matsuzawa, and N. Aoyama. From conceptual models to simulation models: Model driven development of agent-based simulations. In *9th Workshop on Economics and Heterogeneous Interacting Agents*, 2004.

[22] 赤松正教, 古川園智樹, 笠井賢紀, 青山希, 井庭崇. 成長するネットワークのシミュレーションとその拡張：世代交代モデルの提案. MPS シンポジウム「複雑系の科学とその応用」, 2004.

[23] 小林慶太 笠井賢紀 赤松正教 井庭崇古川園智樹. 社会ネットワークの形成過程シミュレーション：マルチエージェント・モデルによる表現と拡張. 情報処理学会 SIG-ICS・人工知能学会 SIG-KBS 合同研究会, 2004.

[24] 井庭崇. 複雑系と進化のモデル・フレームワーク. 西部忠（編）, 進化経済学のフロンティア. 日本評論社, 2004.

[25] 井庭崇, 松澤芳昭, 津屋隆之介. Boxed Economy 基礎モデルに基づく家庭用 VTR の規格競争シミュレーションの作成. 第 7 回進化経済学会, 2003.

[26] 山田悠, 井庭崇. 制限値幅が市場効率性に与える影響の分析: 人工市場アプローチによる分析. 第 7 回進化経済学会, 2003.

[27] R. Tsuya, N. Sato, T. Iba, and Y. Takefuji. Analysis on the factor of price volatility in deregulated electric power market. In *2nd. International Conference of the European Social Simulation Association*, 2004.

[28] D.A. ノーマン. 人を賢くする道具：ソフト・テクノロジーの心理学. 新曜社, 1996.

[29] 佐伯胖. 新・コンピュータと教育. 岩波新書, 1997.